

计算机学院专业必修课

计算机组成

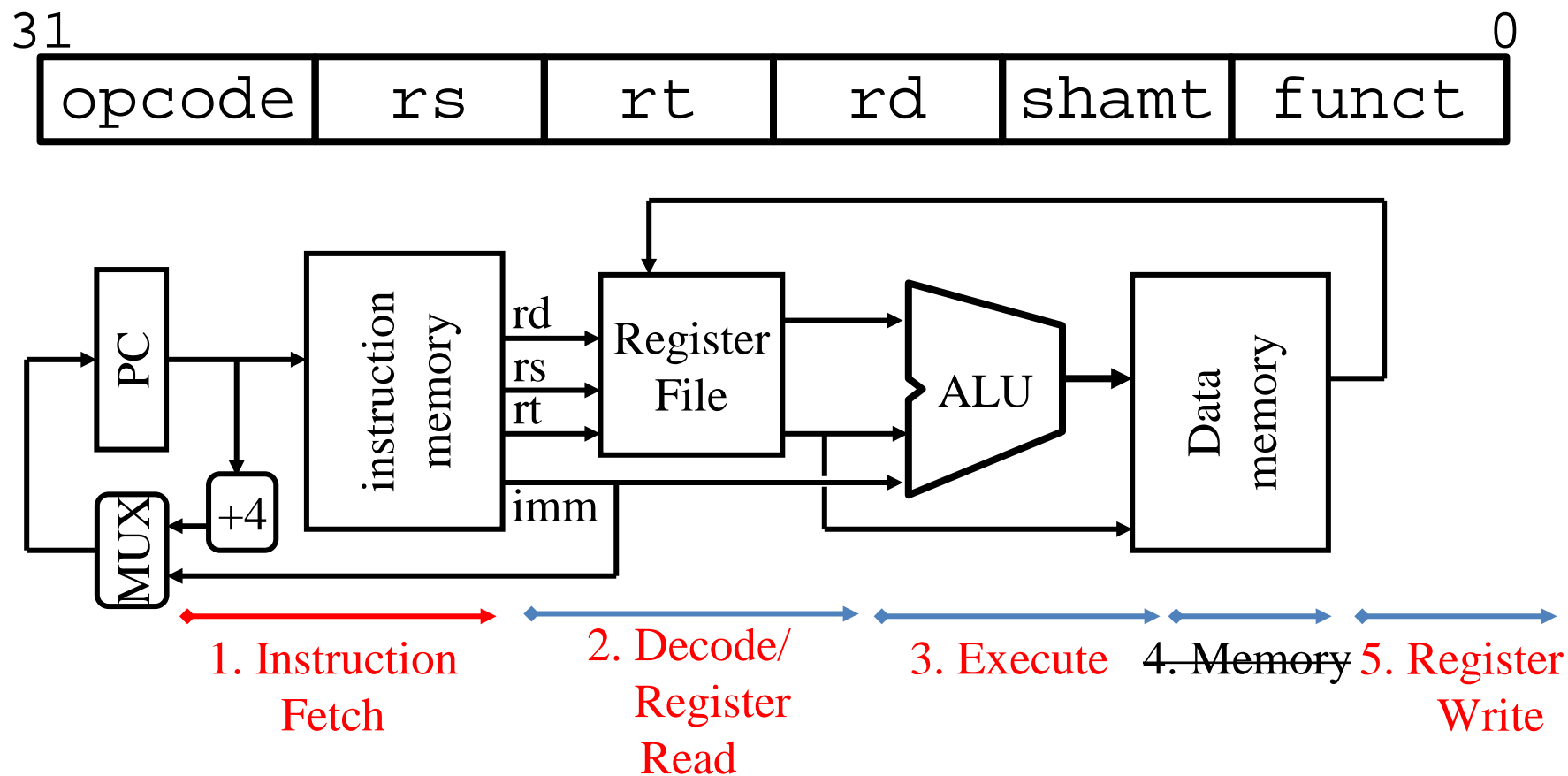
多周期数据通路

高小鹏

北京航空航天大学计算机学院
系统结构研究所

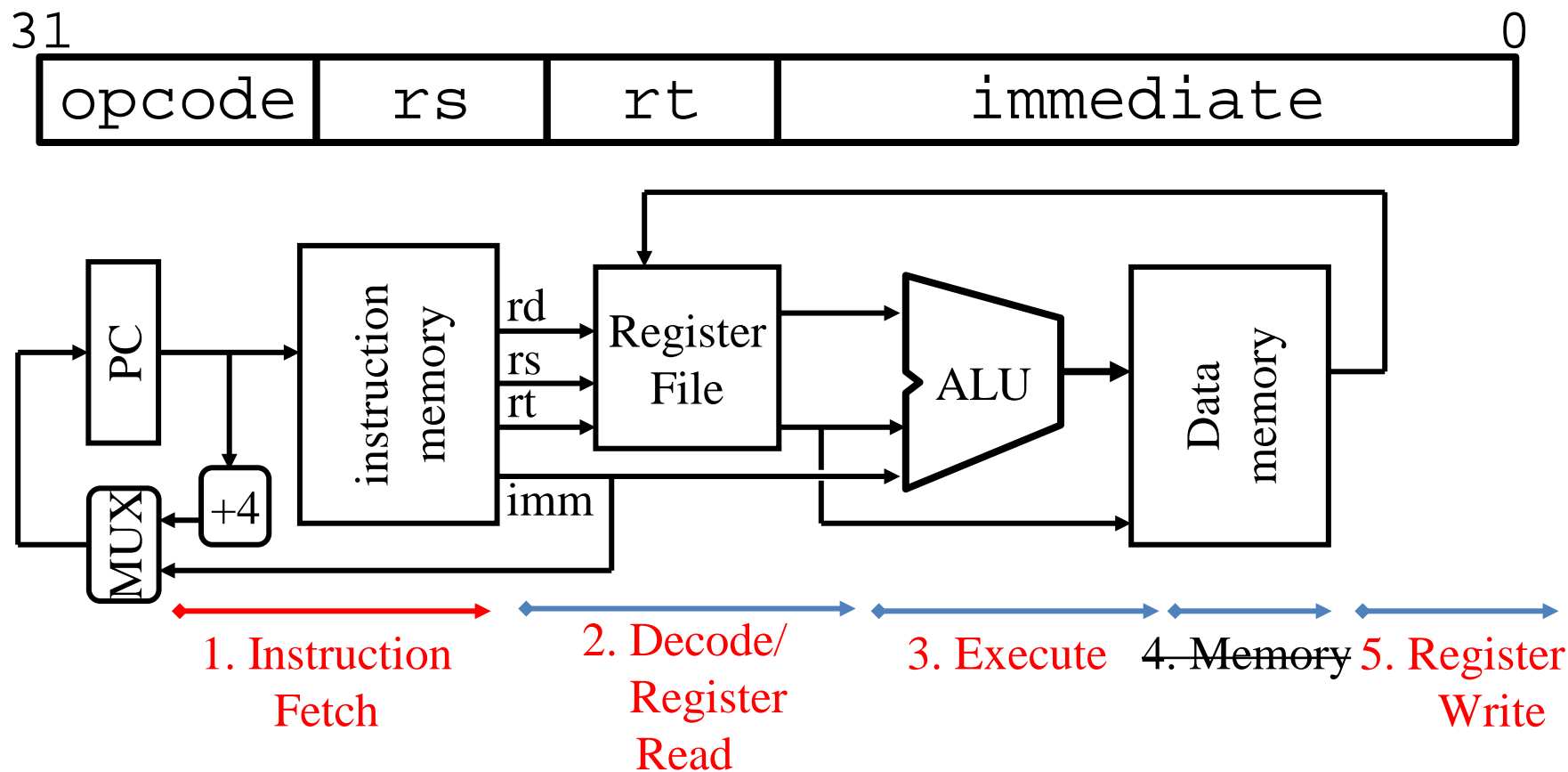
运算类指令执行分析

- 指令：ADD、SUB、OR。。
- 需求： $R[rd] \leftarrow R[rs] \text{ op } R[rt]$
- 过程：取指、译码/读寄存器、执行、访存、回写



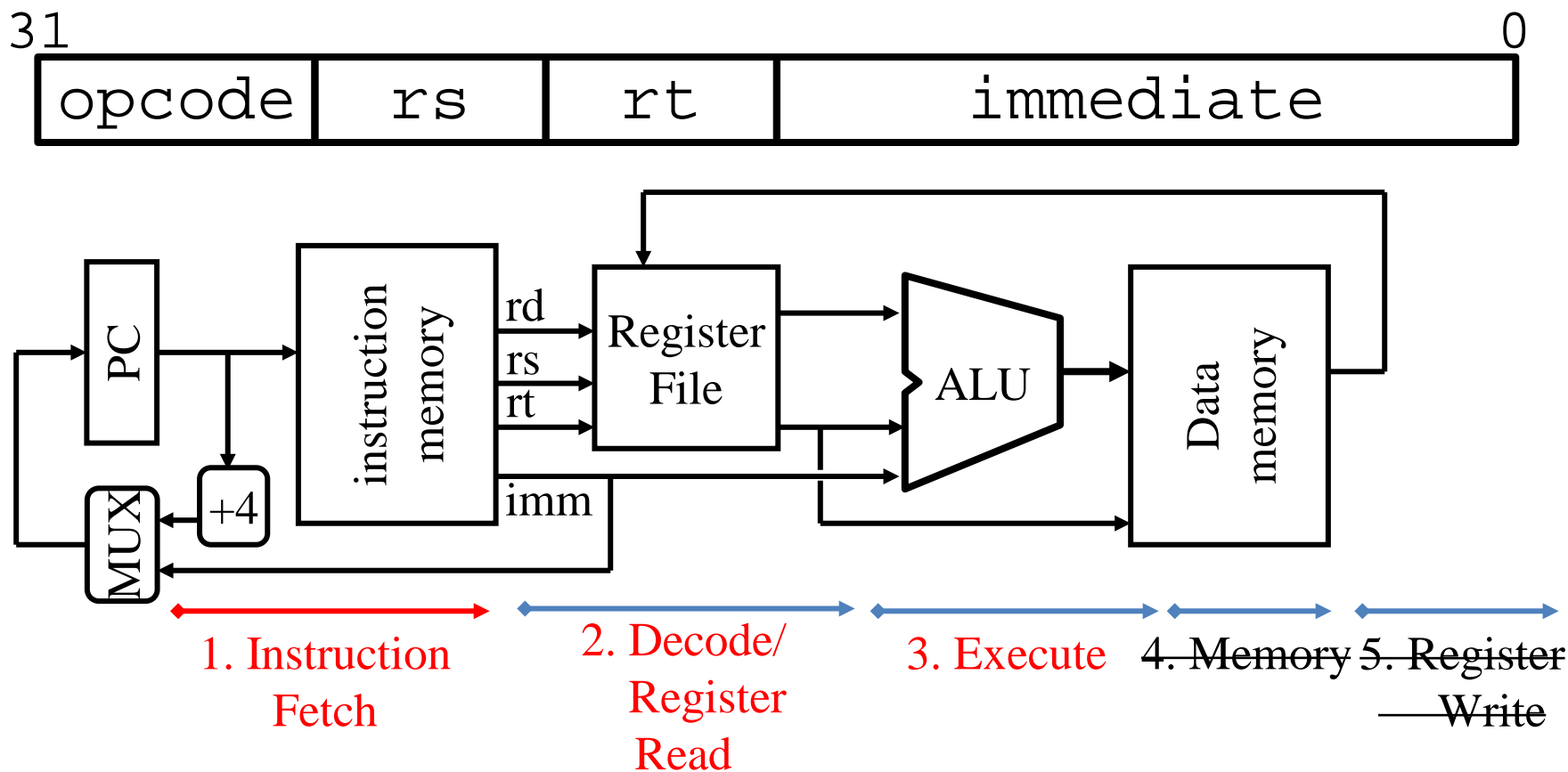
运算类指令执行分析

- 指令：ADDI、SUBI、ORI。。
- 需求： $R[rt] \leftarrow R[rs] \text{ op Ext(Imm)}$
- 过程：取指、译码/读寄存器、执行、访存、回写



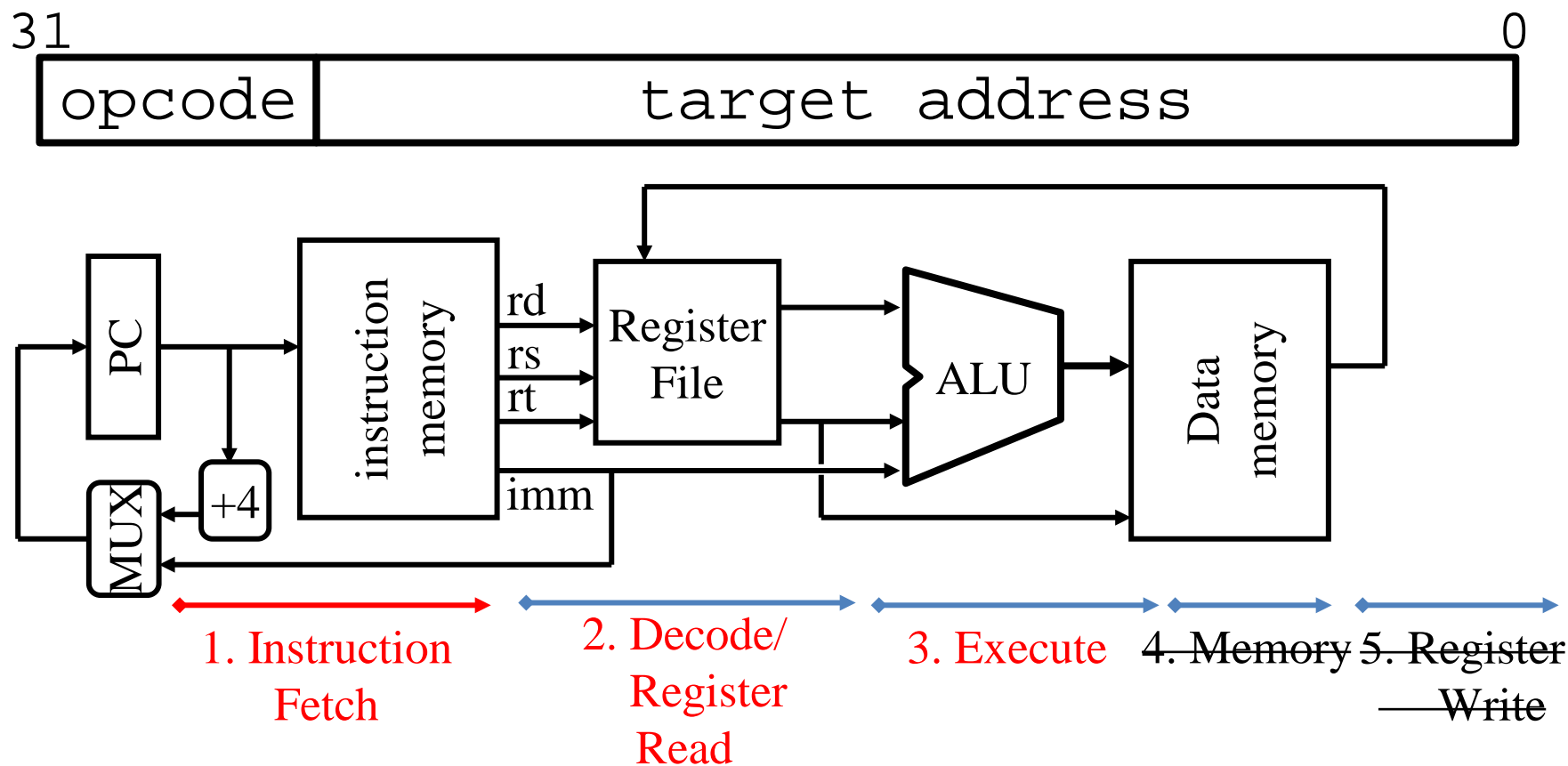
分支类指令执行分析

- 指令：BEQ、...
- 需求：PC \leftarrow 条件? PC + Ext(Imm) : PC + 4
- 过程：取指、译码/读寄存器、执行、访存、回写



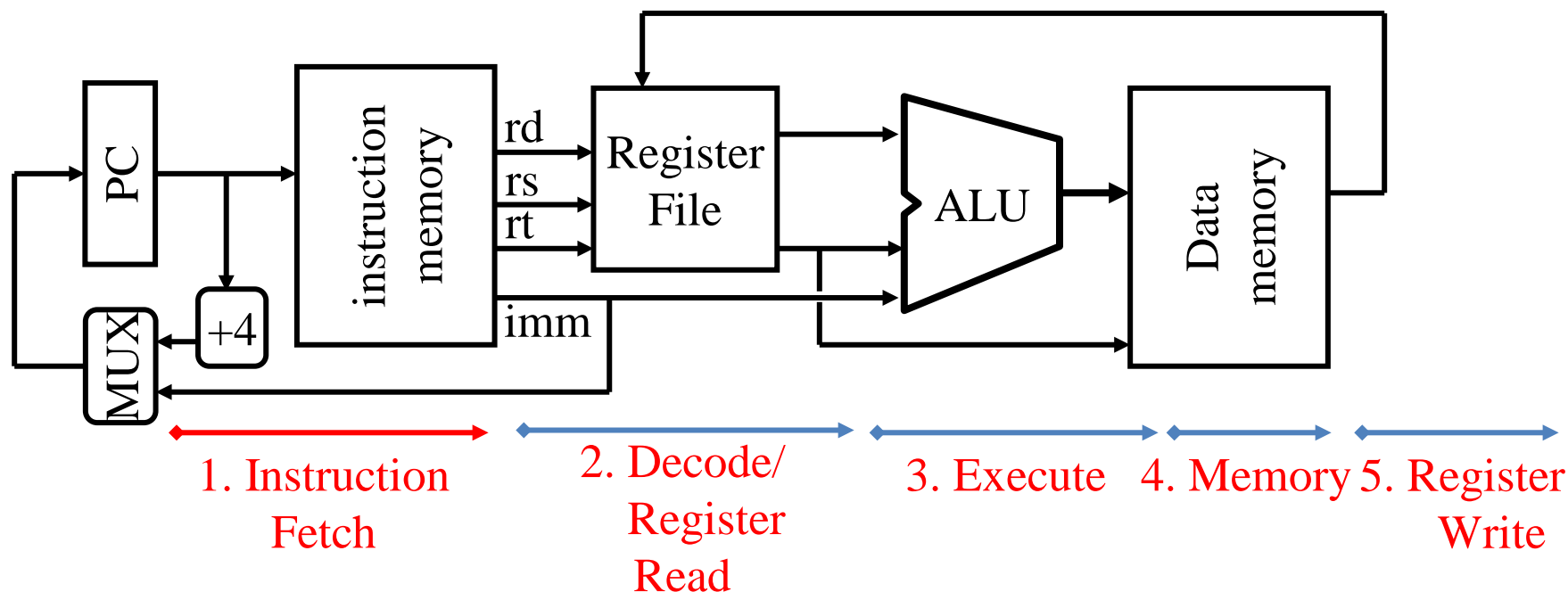
跳转指令执行分析

- 指令：J
- 需求： $PC \leftarrow PC[31:28] \parallel \text{target_address} \parallel 00$
- 过程：取指、译码/读寄存器、执行、访存、回写



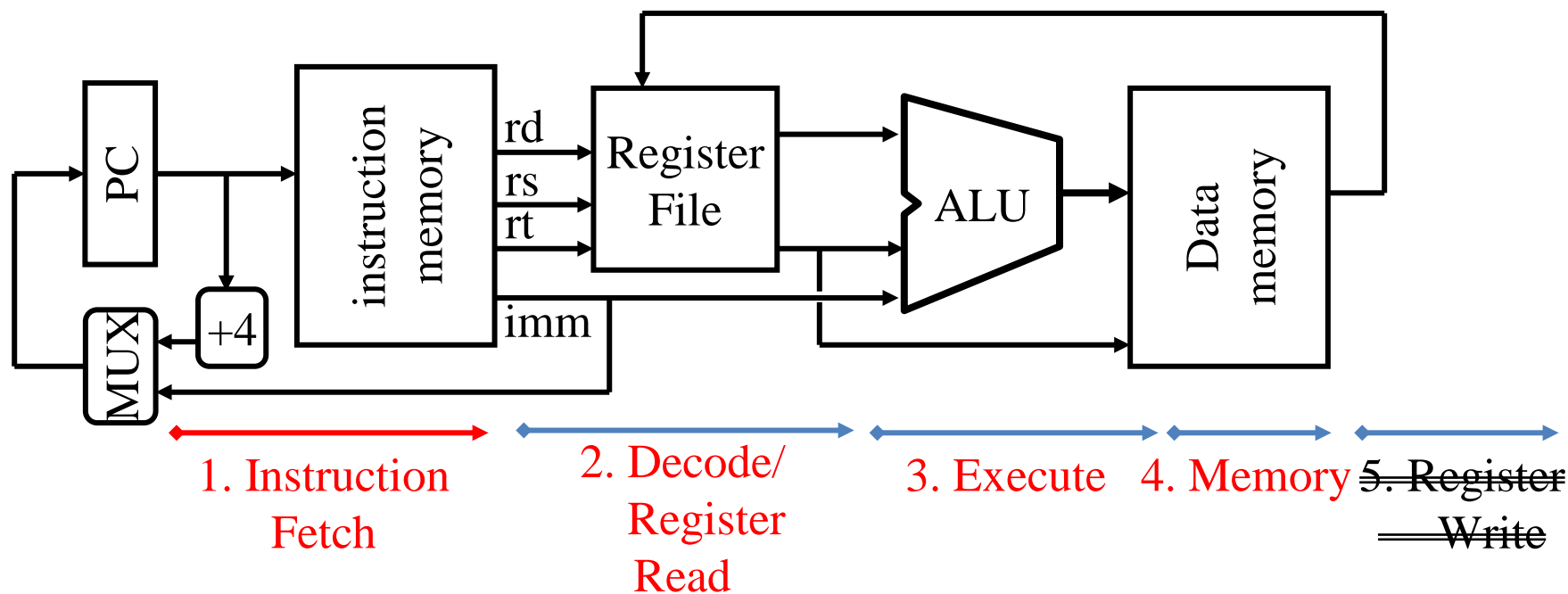
读存储指令执行分析

- 指令：LW
- 需求： $\text{GPR}[\text{rt}] \leftarrow \text{memory}[\text{GPR}[\text{base}] + \text{offset}]$
- 过程：取指、译码/读寄存器、执行、访存、回写



写存储指令执行分析

- 指令：SW
- 需求： $\text{memory}[\text{GPR}[\text{base}] + \text{offset}] \leftarrow \text{GPR}[\text{rt}]$
- 过程：取指、译码/读寄存器、执行、访存、~~回写~~



指令执行过程分析

■ 2个现象

- 现象1：不同指令的执行过程不同
- 现象2：所有指令都有前3个阶段

■ 结论

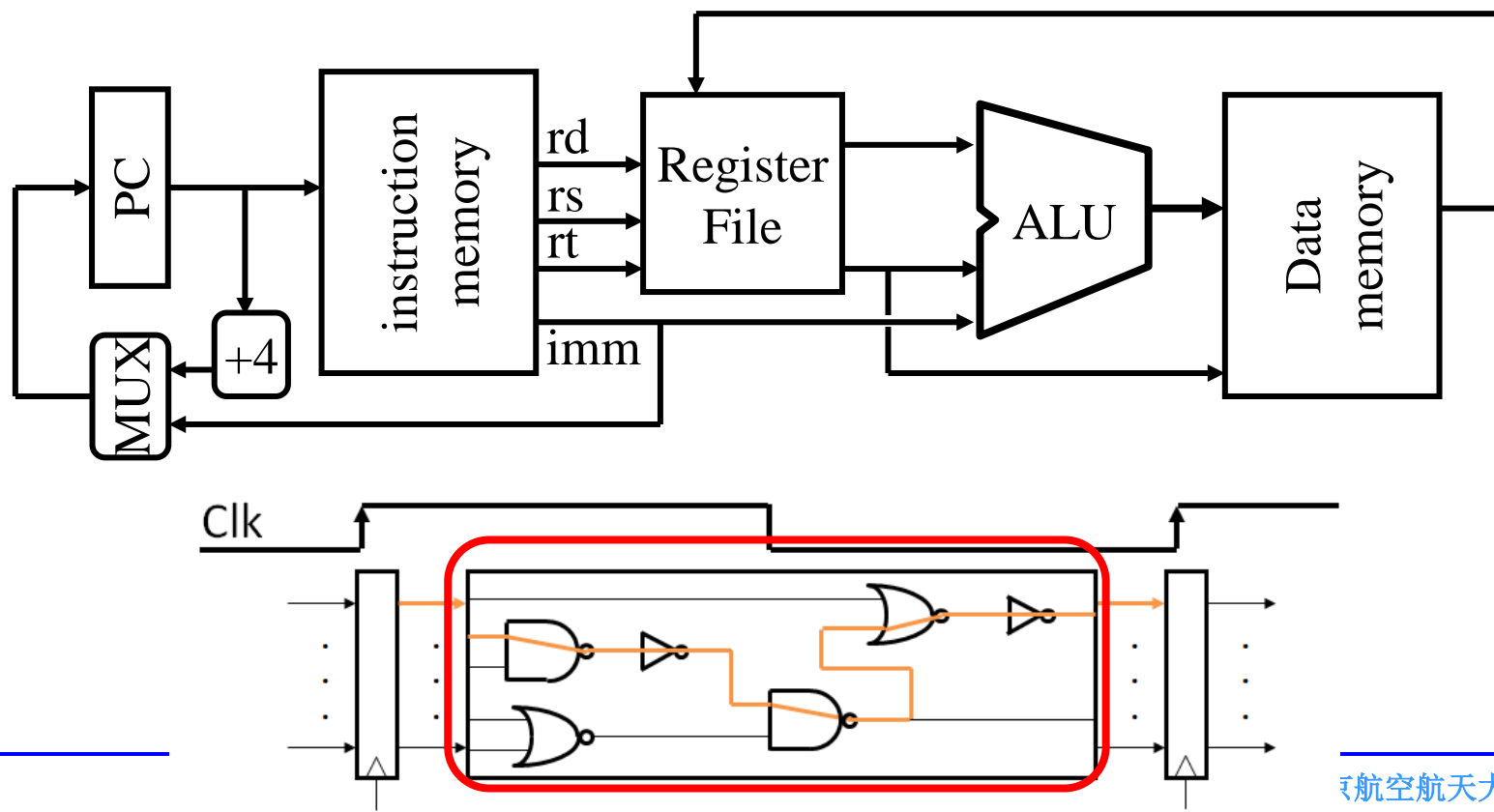
- 多周期数据通路：分割为5个阶段有助于性能提高
- 多周期控制器：需要针对多个阶段控制的控制器

	IF	DCD/RF	EXE	MEM	WB
计算	✓	✓	✓		✓
分支	✓	✓	✓		
跳转	✓	✓	✓		
读存储	✓	✓	✓	✓	✓
写存储	✓	✓	✓	✓	

回顾：单周期数据通路

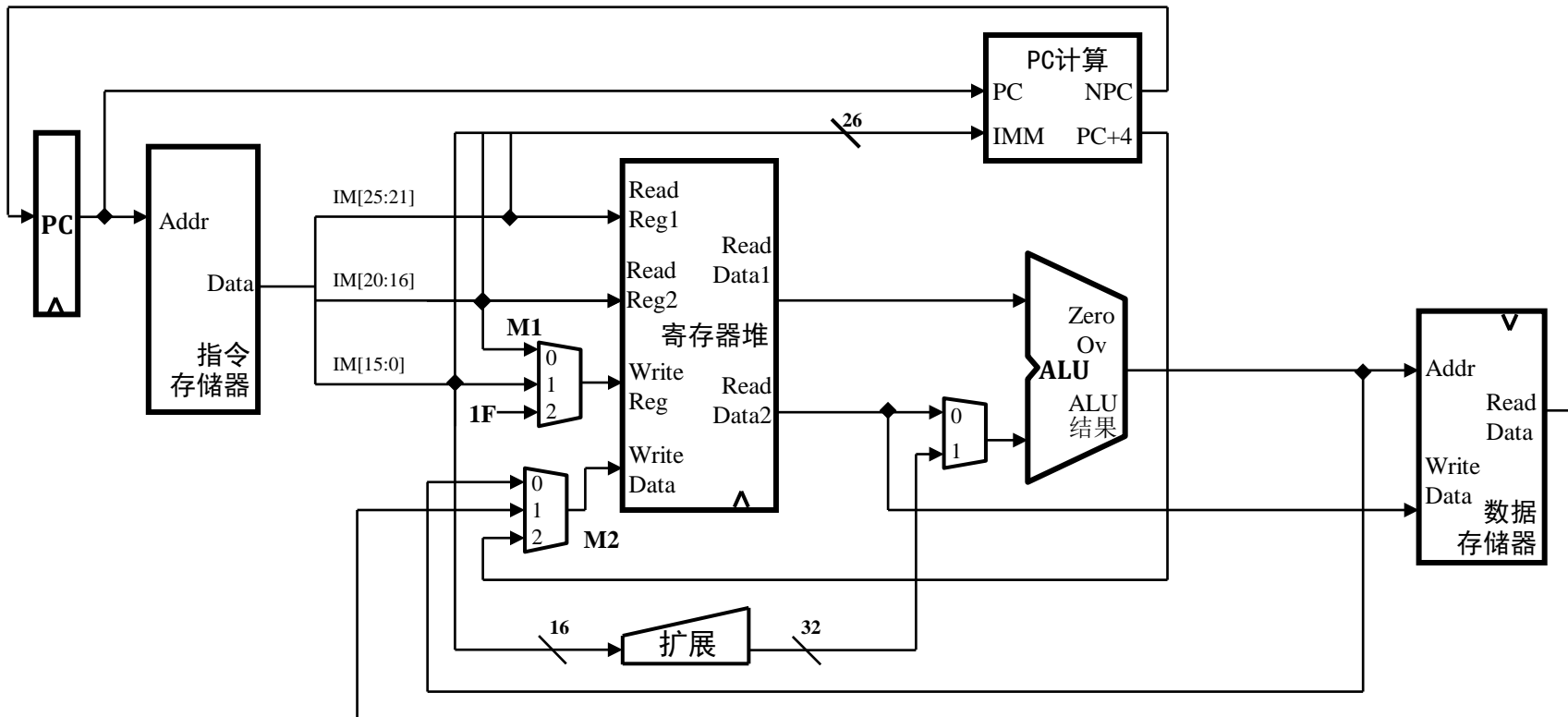
■ 单周期数据通路模型

- PC → 组合逻辑 → 寄存器文件
- 组合逻辑：实现了5个阶段的功能
- 由于关键路径的存在，每条指令延迟均相同
- 无法利用不同指令具有不同执行需求的潜在特性

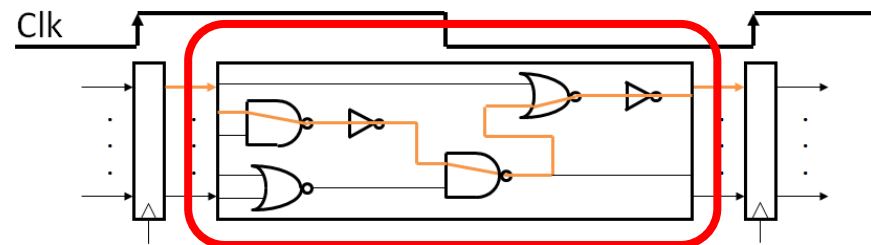


回顾：单周期数据通路

ADDU
SUBU
ORI
LW
SW
BEQ
JAL



- PC、NPC计算单元
- 指令存储器、数据存储器
- 寄存器文件
- 运算单元、扩展单元
- MUX

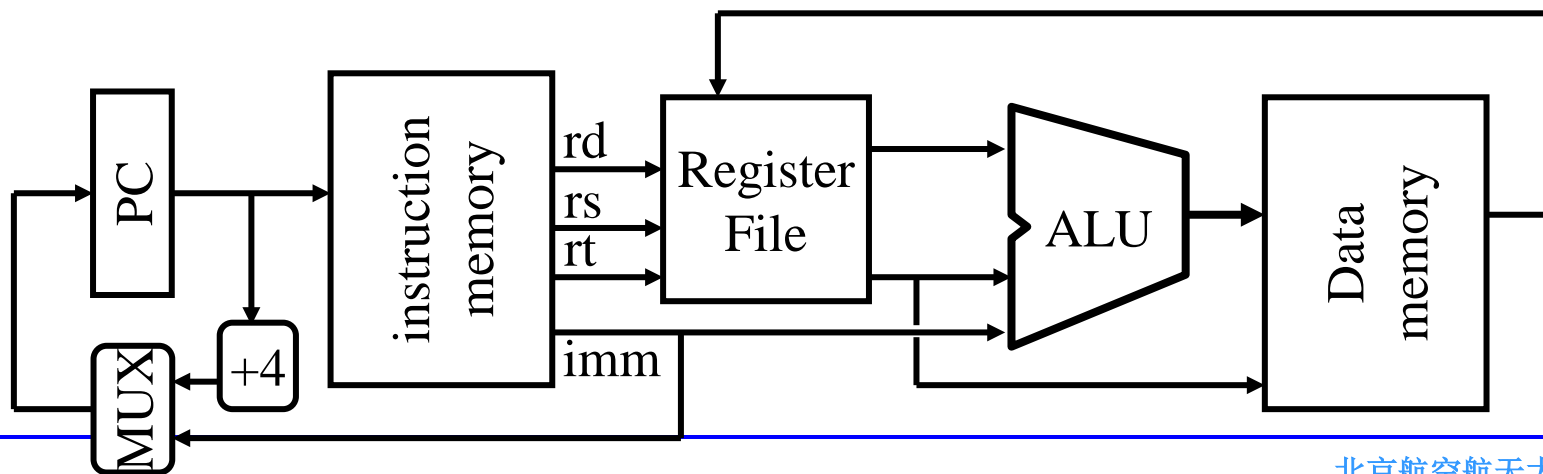


多周期数据通路设计的基本思路

- 目标：不同指令不同执行时间
- 指令不同→执行路径不同
 - R：取指、译码/读寄存器、执行、访存、回写

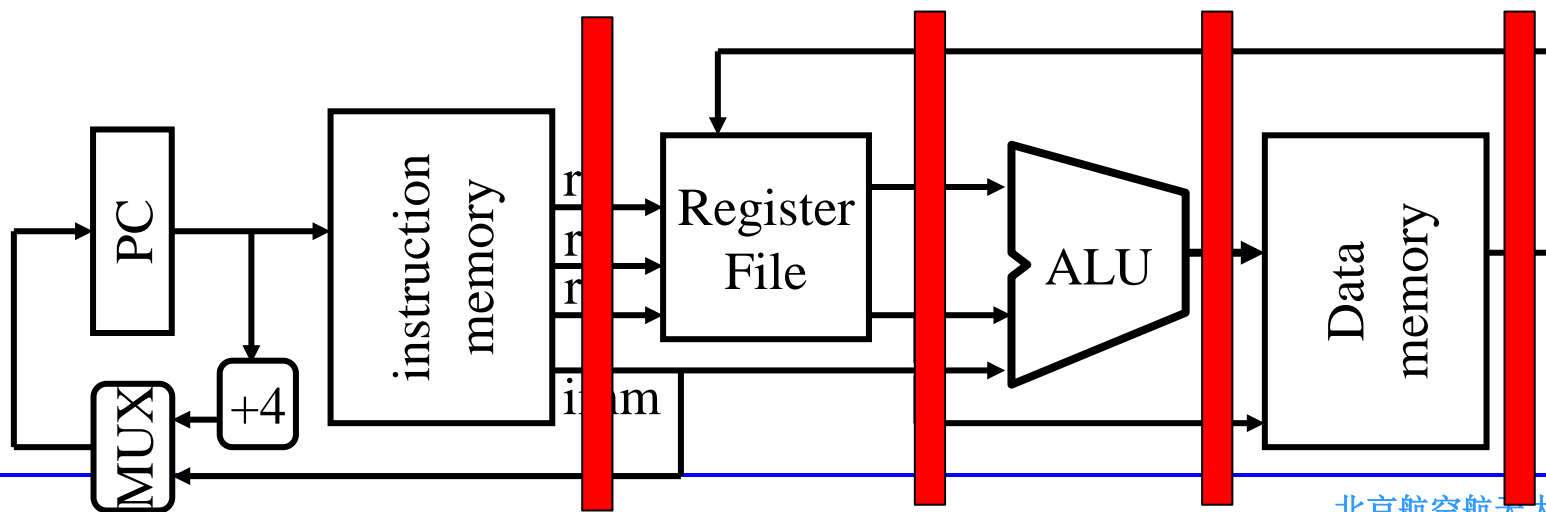
多周期数据通路构思

- 单周期数据通路：只有1层寄存器与组合逻辑
 - 可以将RF与PC看成为同一阶段
 - 组合逻辑：实现了5个阶段的功能
- 多周期数据通路设计构思
 - 在组合逻辑中插入寄存器，切分数据通路
 - 单一组合逻辑被切分为若干小组合逻辑
 - 单一大延迟变为多个分段小延迟



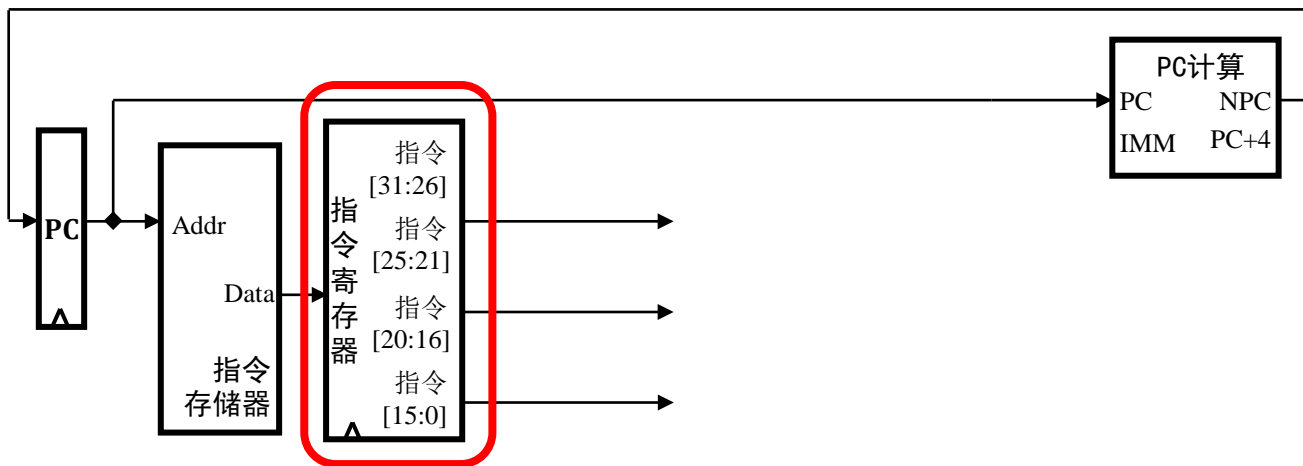
多周期数据通路构思

- 多周期数据通路设计构思
 - 在组合逻辑中插入寄存器，切分数据通路
 - 大组合逻辑被切分为若干小组合逻辑
 - 大延迟变为多个分段小延迟
 - 不同指令执行占用不同的功能单元
 - ◆ 不必5个环节都走完



ADDU指令的多周期数据通路：IF阶段

ADDU
SUBU
LW
SW
ORI
LUI
BEQ
JAL

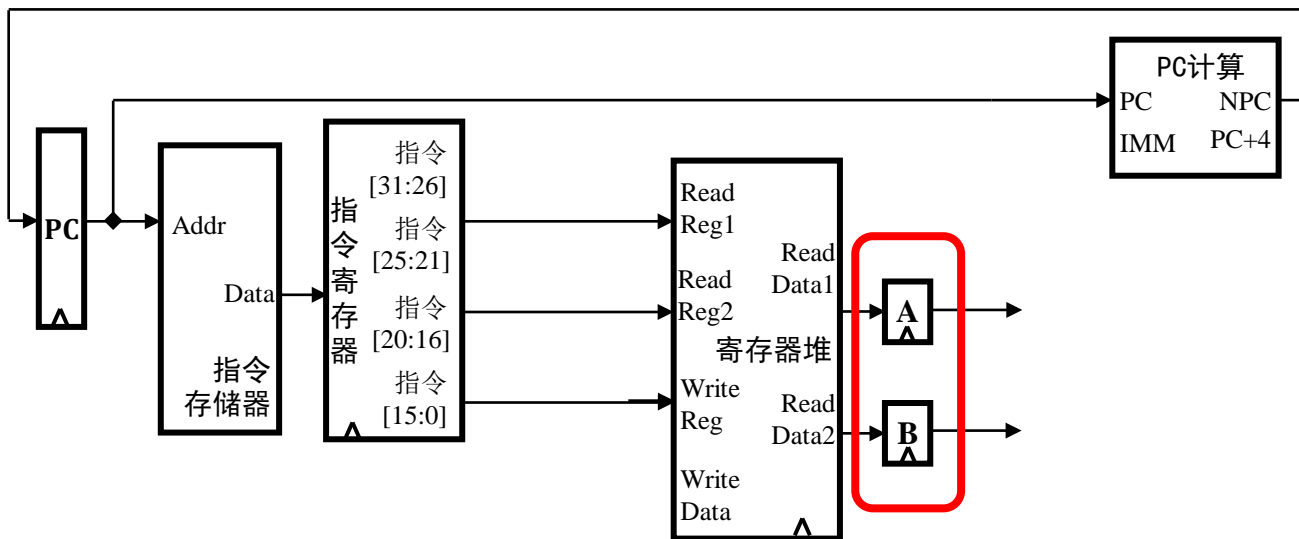


- 加载指令：读取IM，写入IR(指令寄存器)
- 更新PC： $PC \leftarrow PC + 4$

TIP: PC需要写使能
⇒ 与单周期不同了!

ADDU指令的多周期数据通路：DCD/**RF**阶段

ADDU
SUBU
LW
SW
ORI
LUI
BEQ
JAL

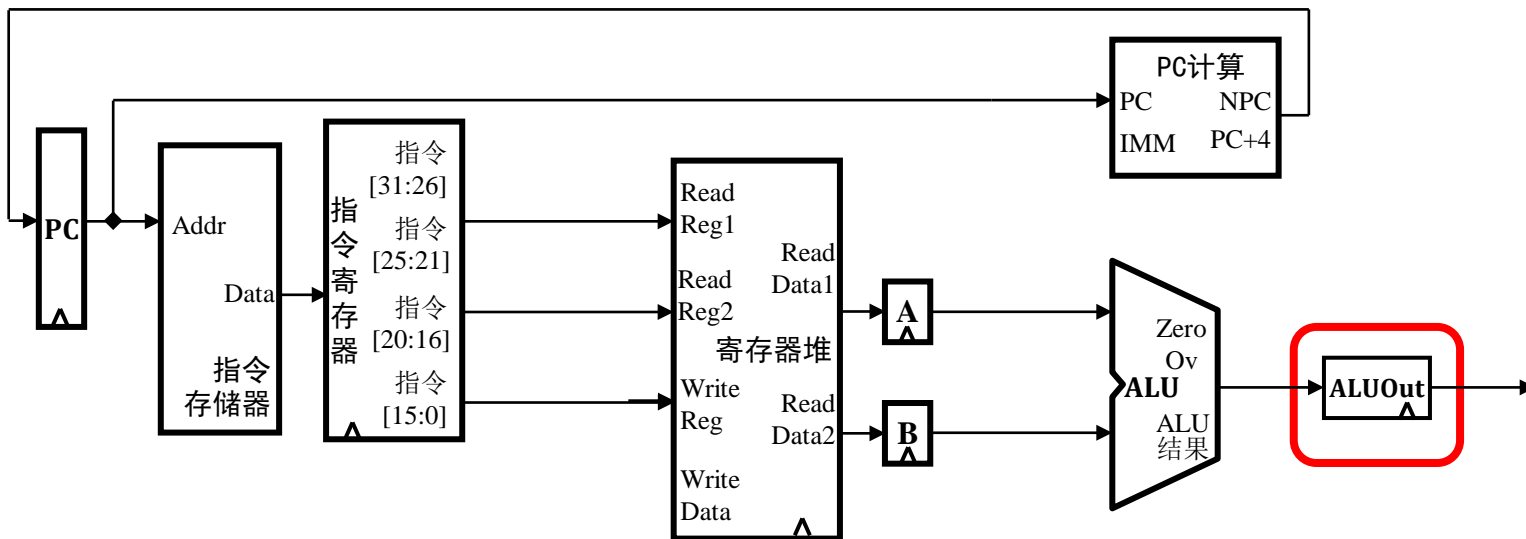


- 读取操作数：GPR→A、B
 - A、B：分别存储寄存器文件的2个输出

TIP：A、B不需要写使能
⇒ 为什么？

ADDU指令的多周期数据通路：EXE阶段

ADDU
SUBU
LW
SW
ORI
LUI
BEQ
JAL



■ 存储计算结果

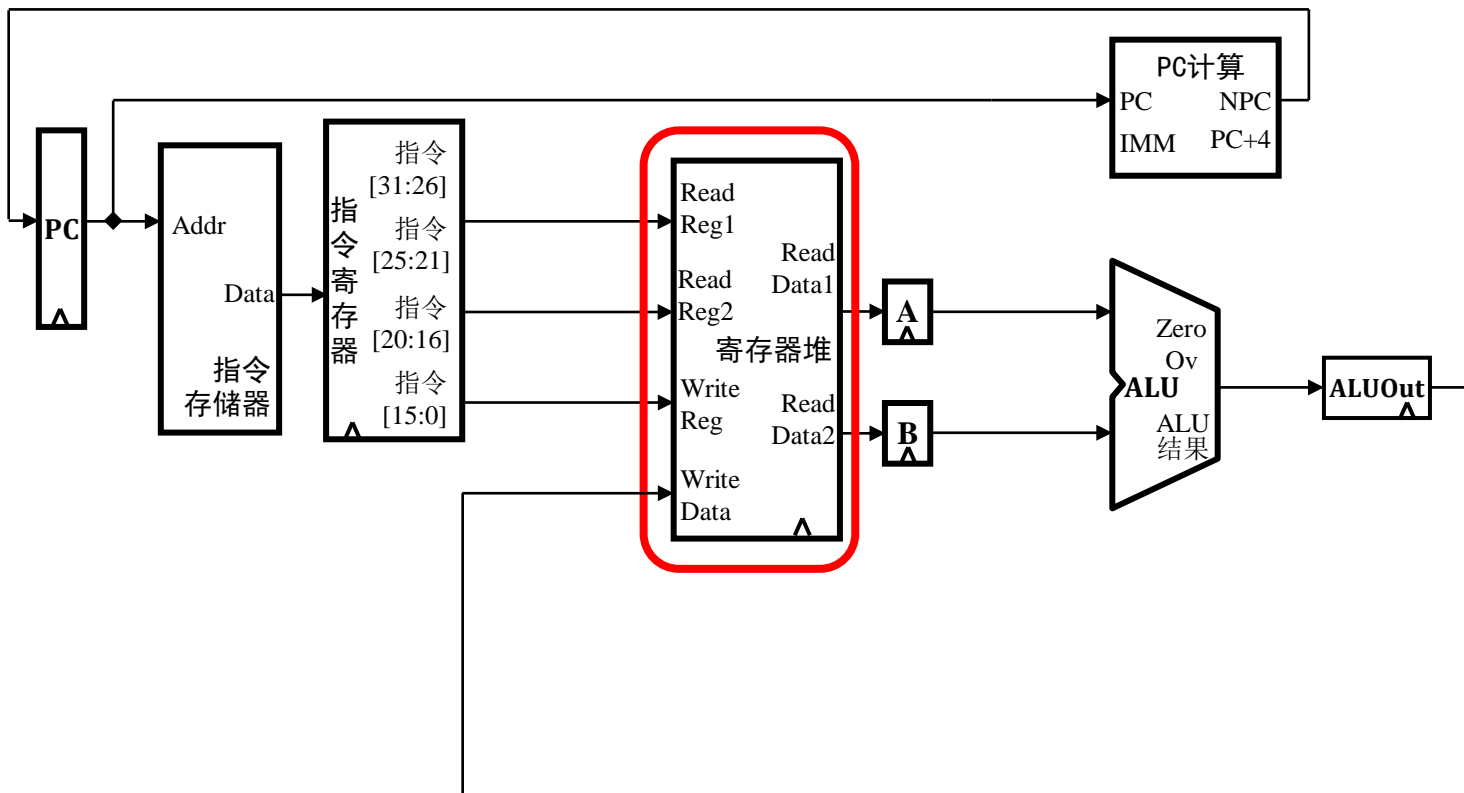
- ALU计算结果存储在ALUOut寄存器

TIP: ALUOut不需要写使能

⇒ 为什么?

ADDU指令的多周期数据通路：WB阶段

ADDU
SUBU
LW
SW
ORI
LUI
BEQ
JAL

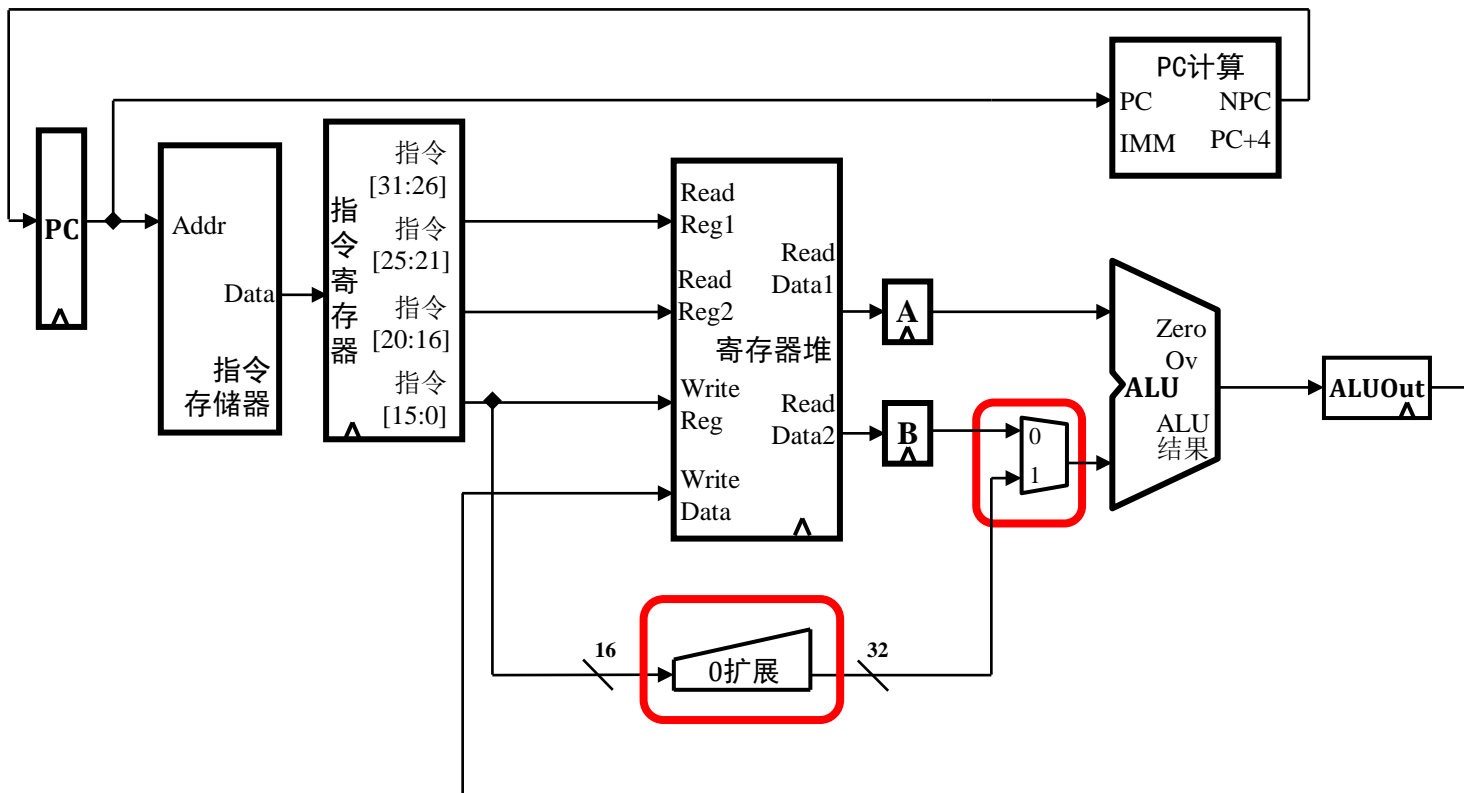


■ 回写寄存器

- ALUOut存储的计算结果写入对应的寄存器

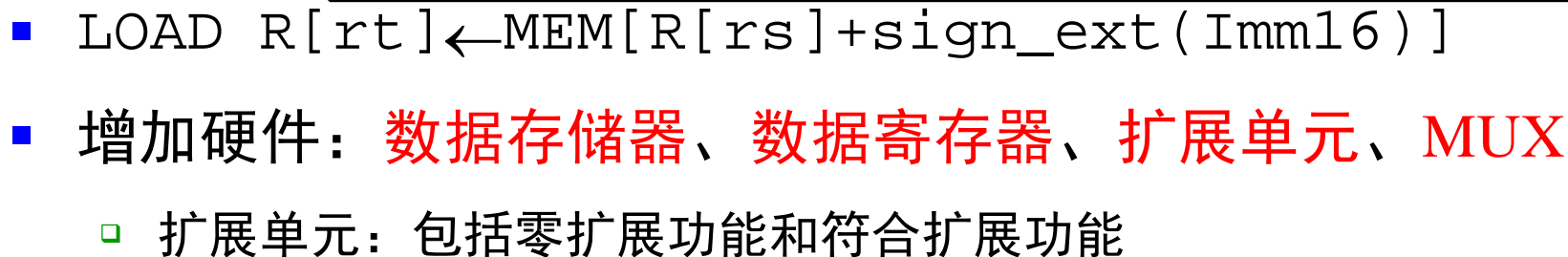
ORI指令的多周期数据通路

ADDU
SUBU
ORI
LW
SW
BEQ
JAL



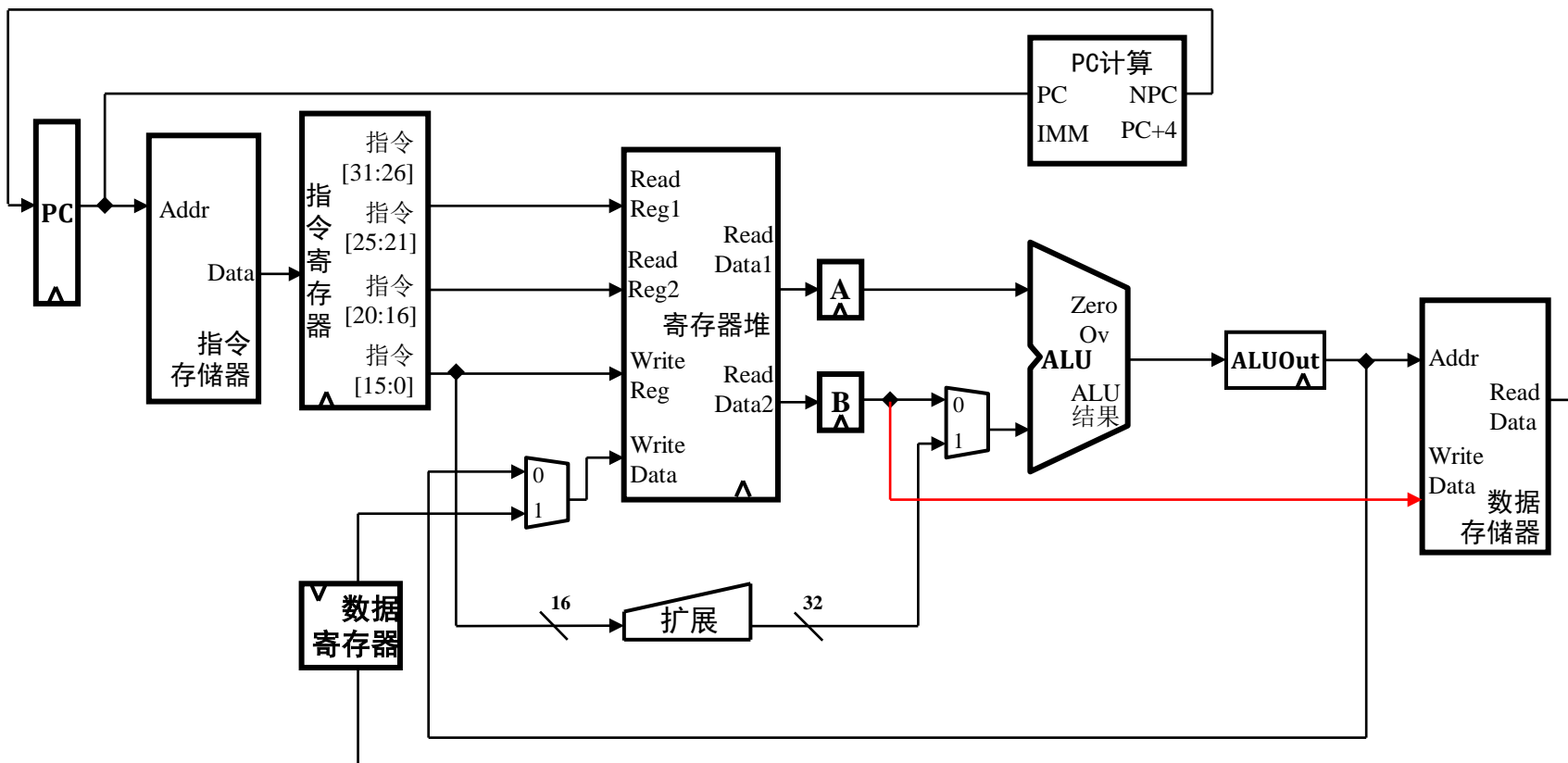
- $\text{ORI } R[rt] \leftarrow R[rs] \mid \text{zero_ext}(\text{Imm16}) ;$
- 增加硬件：**零扩展单元**、**MUX**

ADDU
SUBU
ORI
LW
SW
BEQ
JAL



SW指令的多周期数据通路

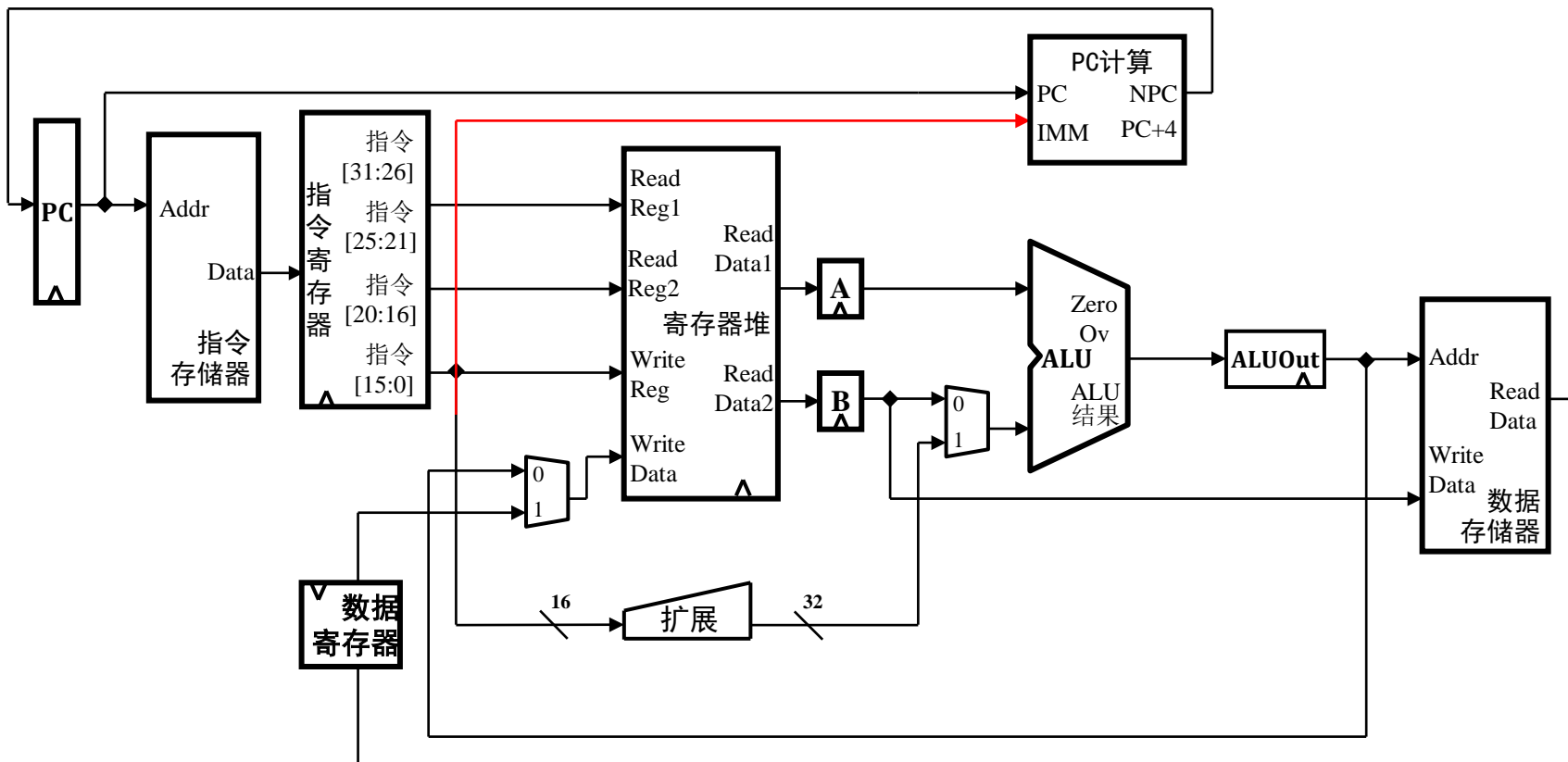
ADDU
SUBU
ORI
LW
SW
BEQ
JAL



- $\text{STORE MEM}[R[\text{rs}] + \text{sign_ext}(\text{Imm16})] \leftarrow R[\text{rt}];$
- 增加硬件：连接线
 - 从B寄存器至数据存储器的

BEQ指令的多周期数据通路

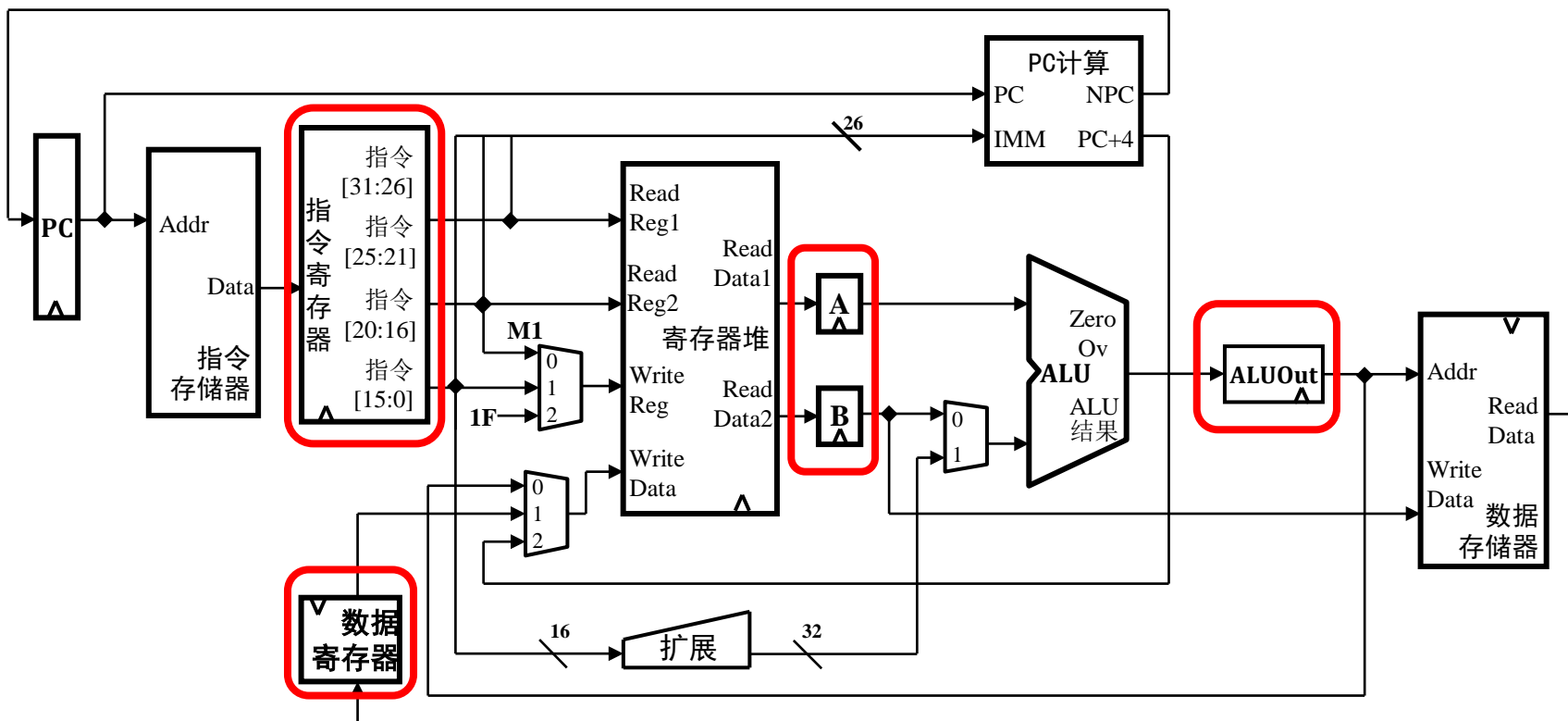
ADDU
SUBU
ORI
LW
SW
BEQ
JAL



- $PC \leftarrow (GPR[rs] = GPR[rt]) ? PC + \text{sign_ext}(imm16) : PC + 4$
- 增加硬件：连接线
 - 从IM[15:0]至PC计算单元

多周期数据通路

ADDU
SUBU
ORI
LW
SW
BEQ
JAL

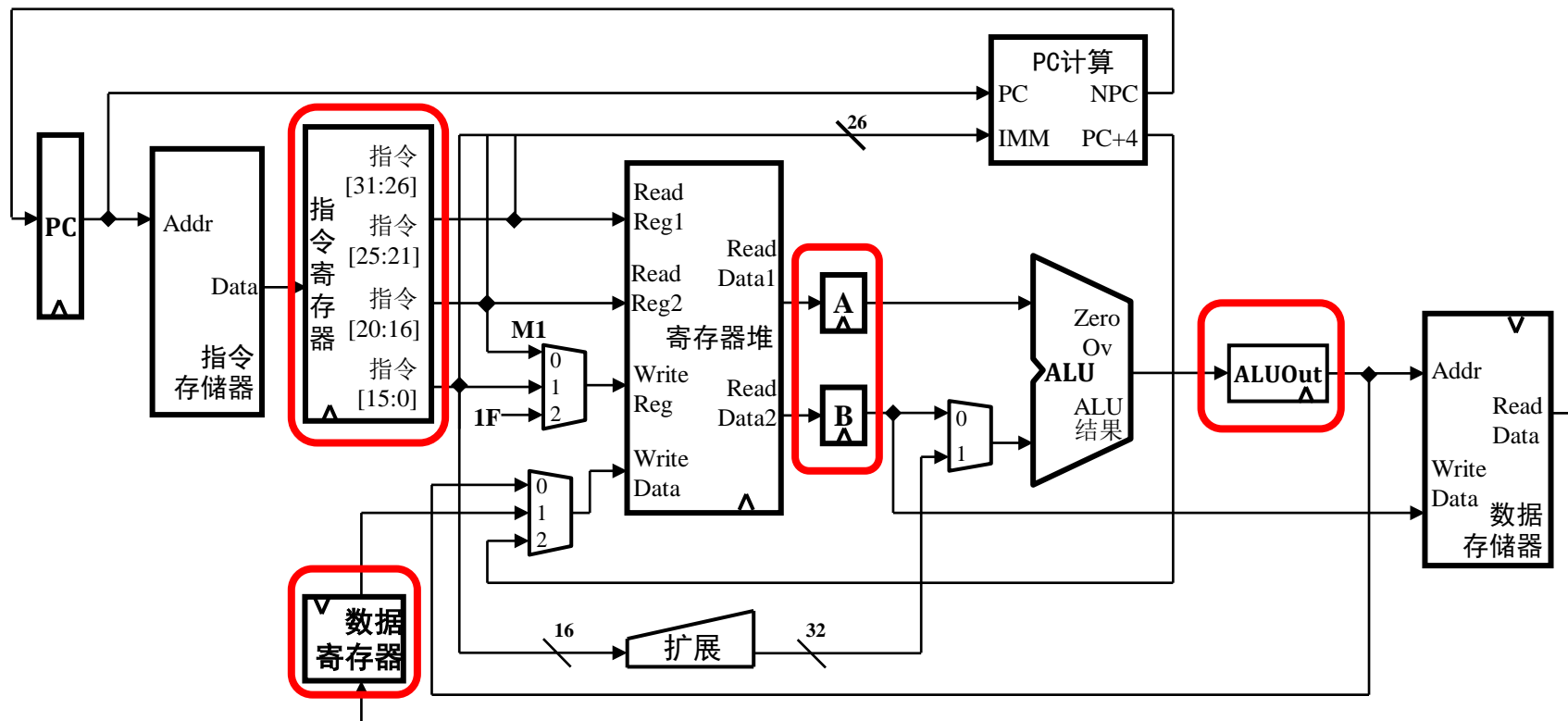


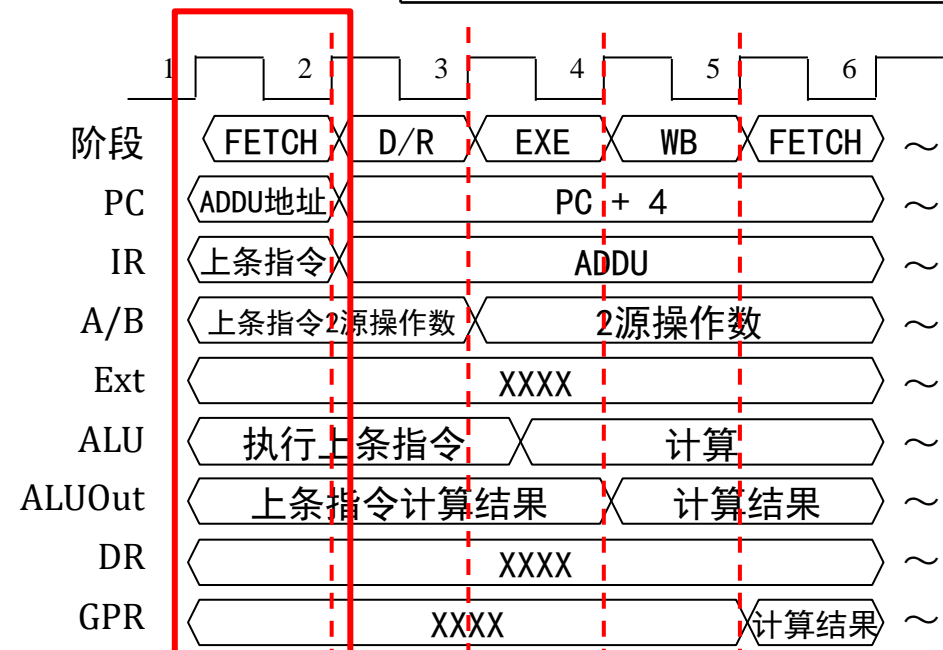
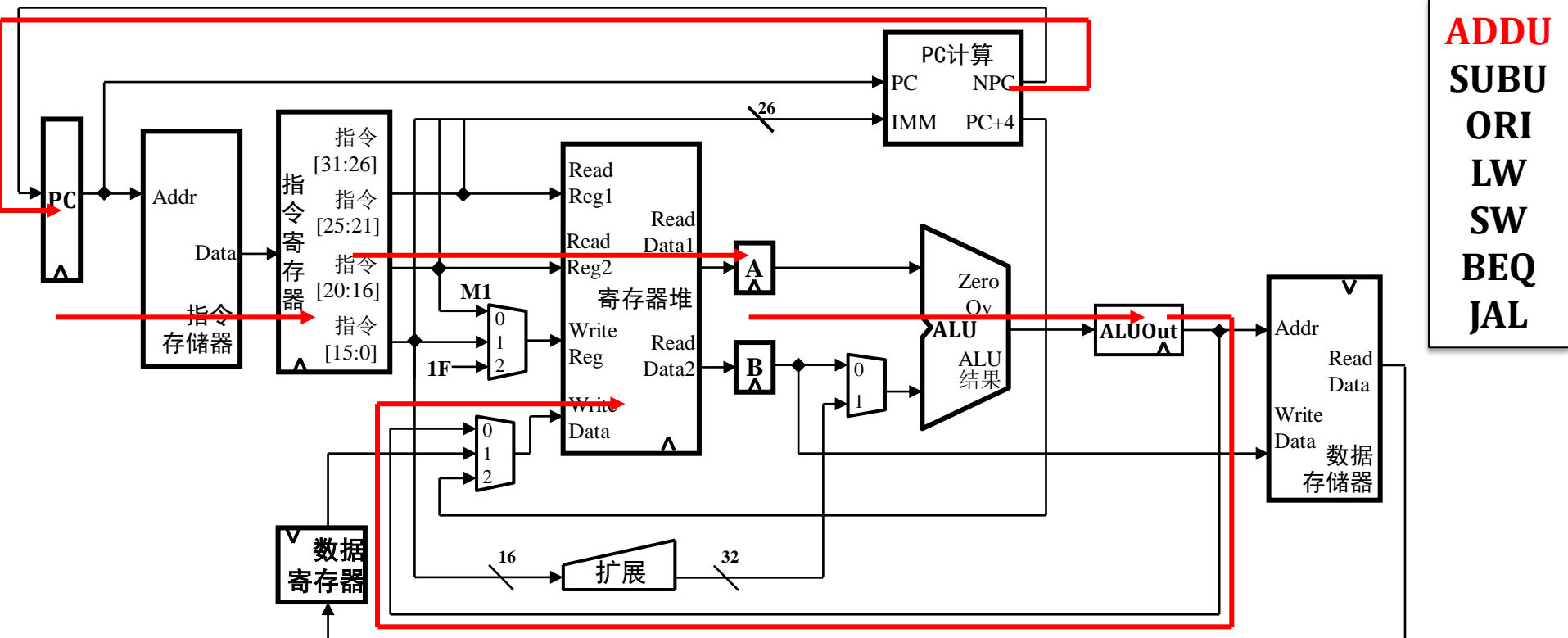
■ 增加硬件

- IR：指令寄存器
- A/B：操作数寄存器
- ALUOut：计算结果寄存器
- DR：数据寄存器

分析的要点

- PC不变?
- IR不变?





执行

ADDU

SUBU

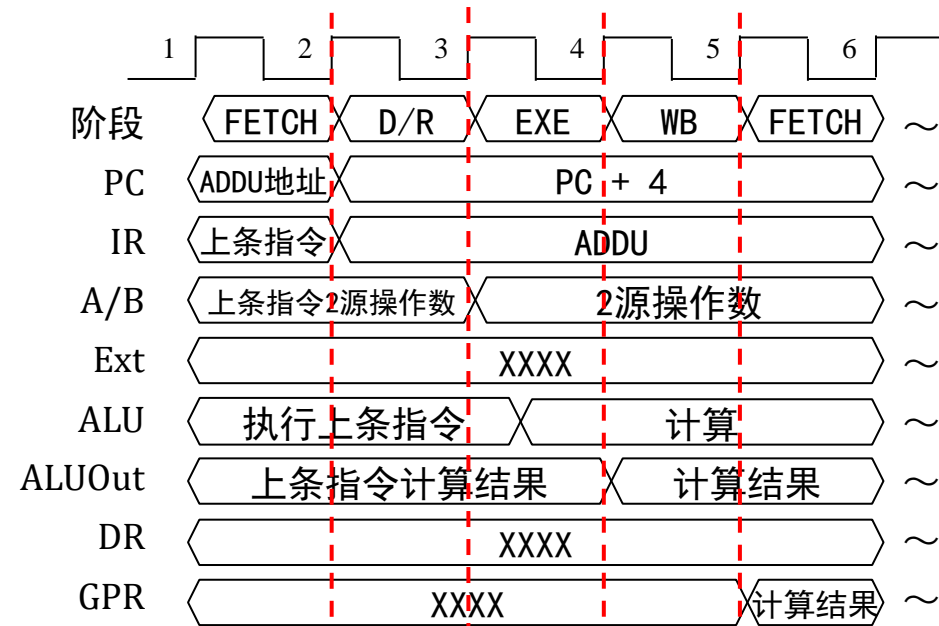
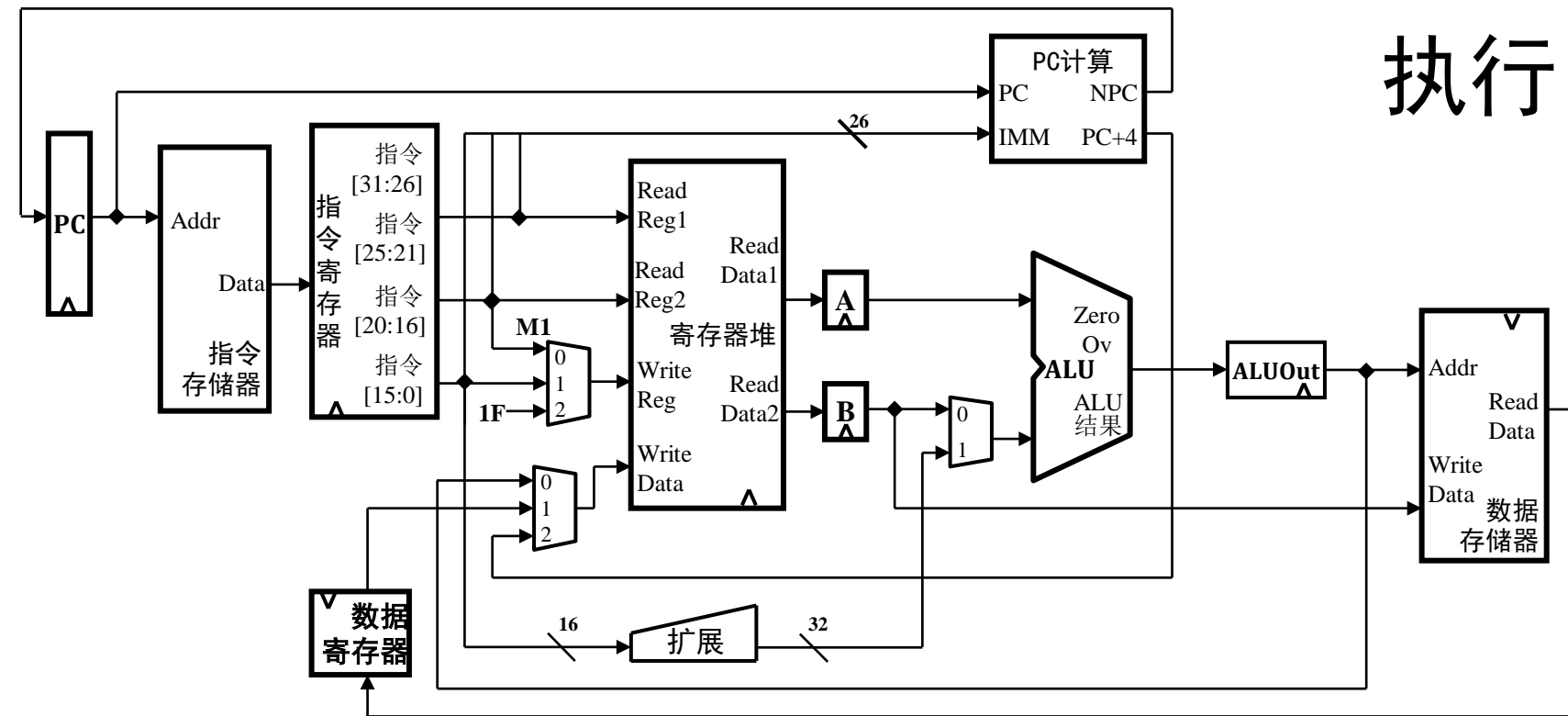
ORI

LW

SW

BEQ

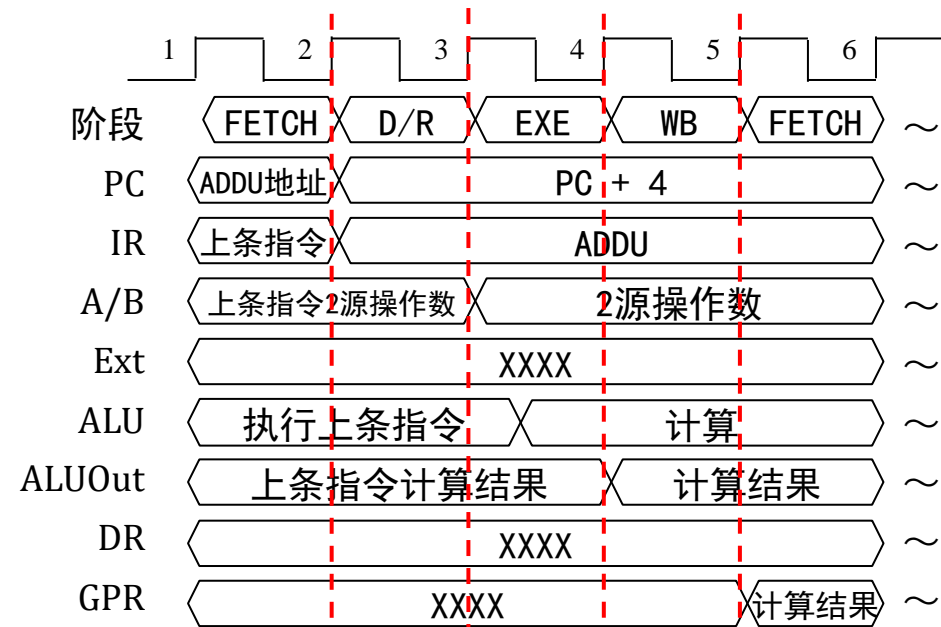
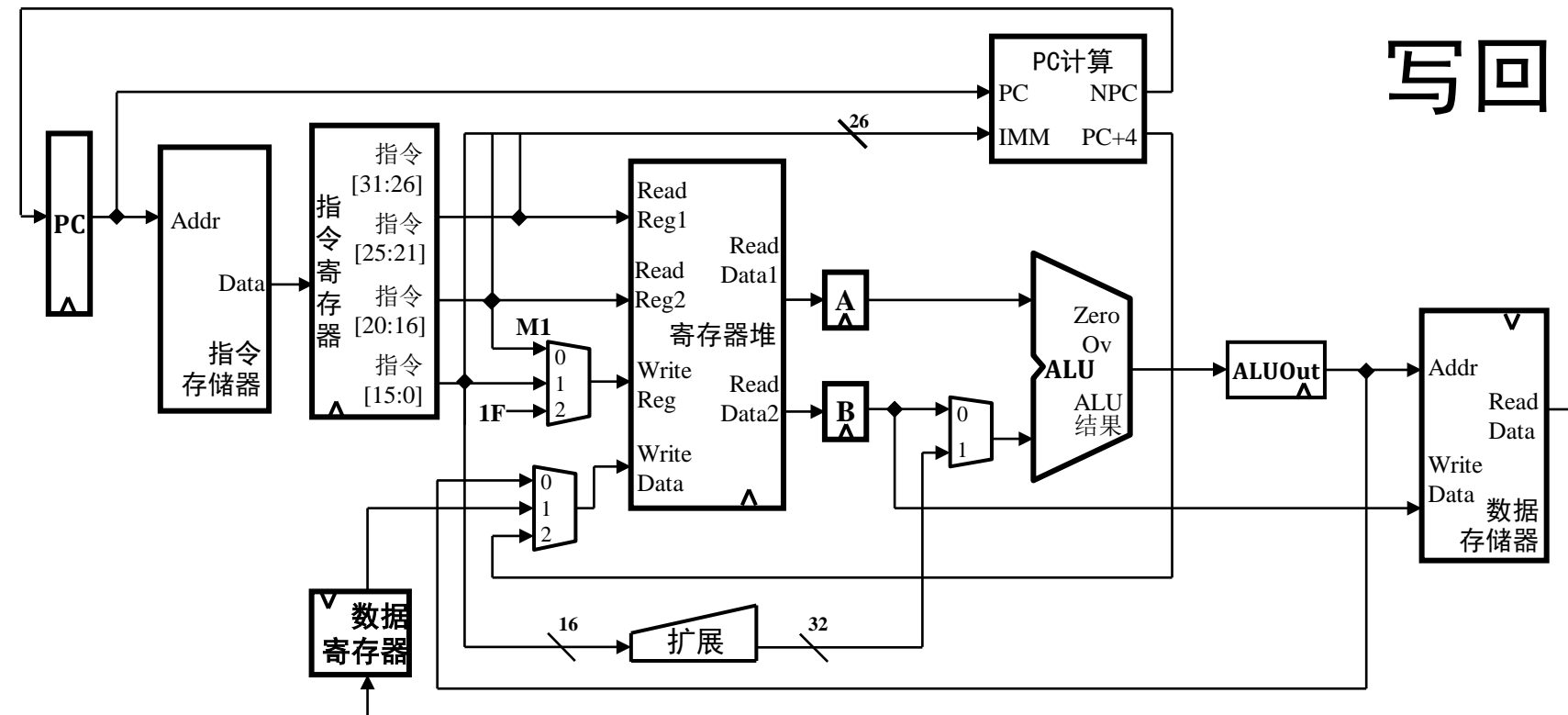
JAL



	FETCH	DCD/RF	EXE	WB	FETCH
PC	ADDU地址	PC+4	PC+4		
IM	输出ADDU	指令[PC+4]	指令[PC+4]		
IR	旧指令	ADDU	ADDU		
A/B	旧操作数	旧操作数	新操作数		
Ext	无效扩展	无效扩展	无效扩展		
ALU	执行前指令	执行前指令	计算		
ALUOut	前结果	前结果	前结果		
DM	无效数据	无效数据	无效数据		
DR	无效数据	无效数据	无效数据		
GPR	旧值输出	新值输出	新值输出		

写回

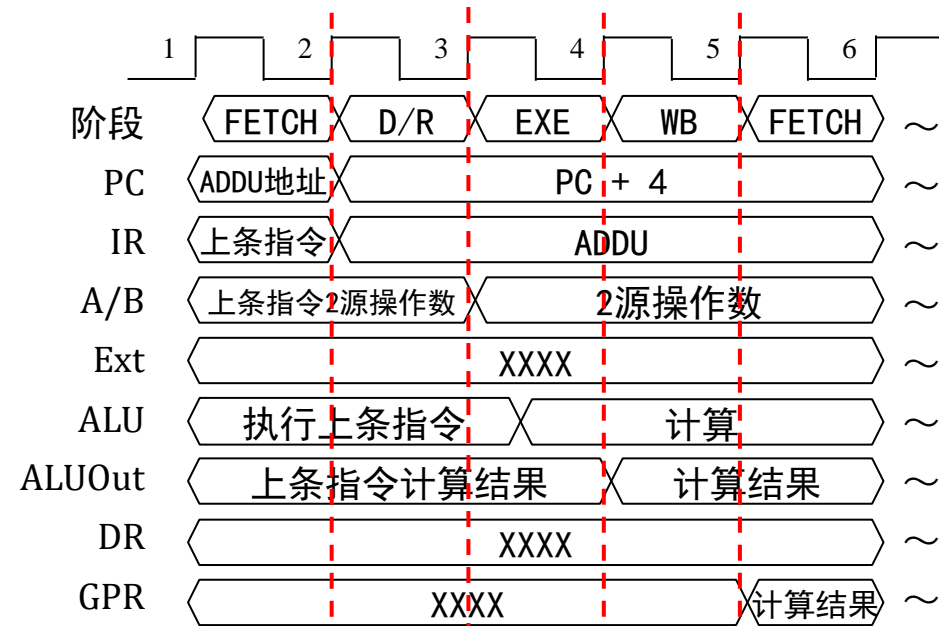
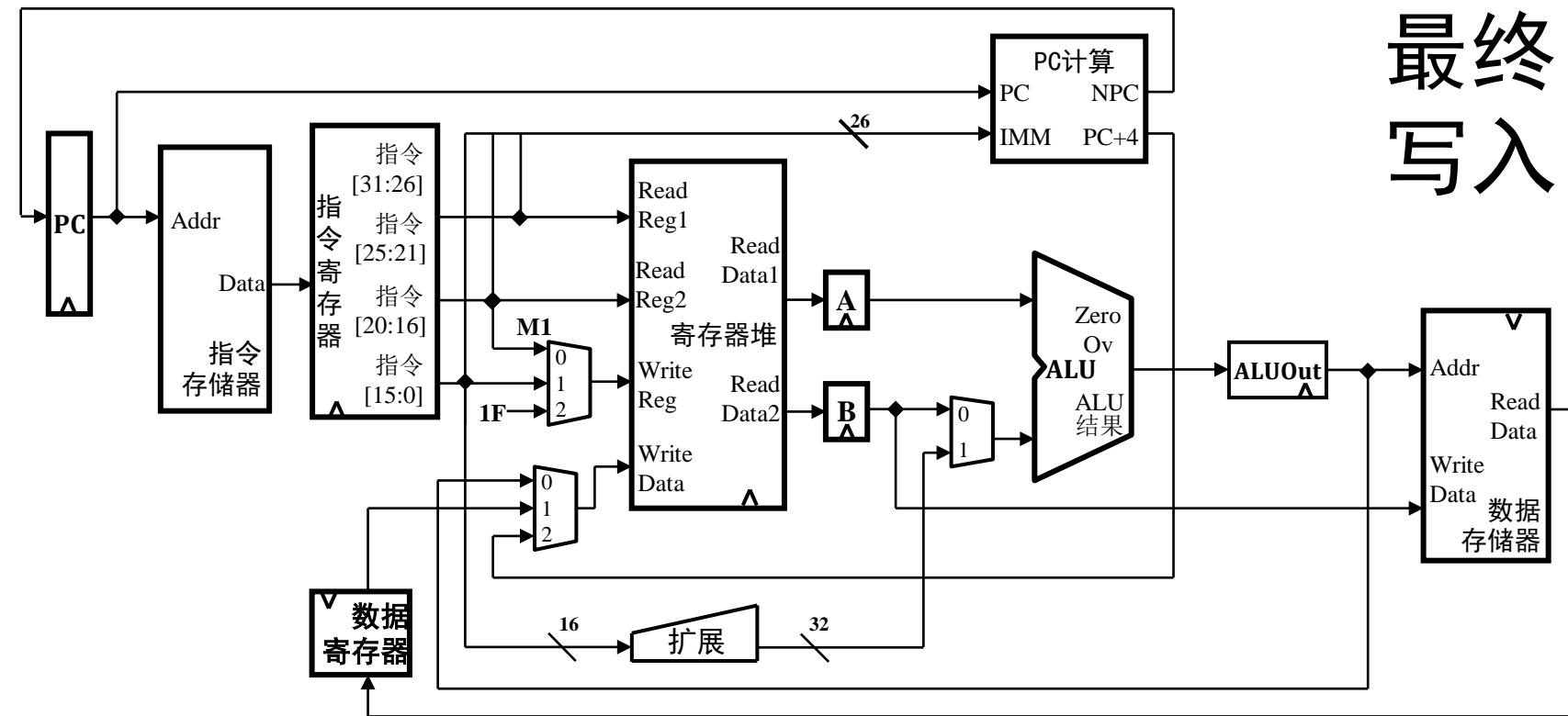
ADDU
SUBU
ORI
LW
SW
BEQ
JAL



	FETCH	DCD/RF	EXE	WB	FETCH
PC	ADDU地址	PC+4	PC+4	PC+4	
IM	输出ADDU	指令[PC+4]	指令[PC+4]	指令[PC+4]	
IR	旧指令	ADDU	ADDU	ADDU	
A/B	旧操作数	旧操作数	新操作数	新操作数	
Ext	无效扩展	无效扩展	无效扩展	无效扩展	
ALU	执行前指令	执行前指令	计算	计算	
ALUOut	前结果	前结果	前结果	计算结果	
DM	无效数据	无效数据	无效数据	无效数据	
DR	无效数据	无效数据	无效数据	无效数据	
GPR	旧值输出	新值输出	新值输出	新值输出	

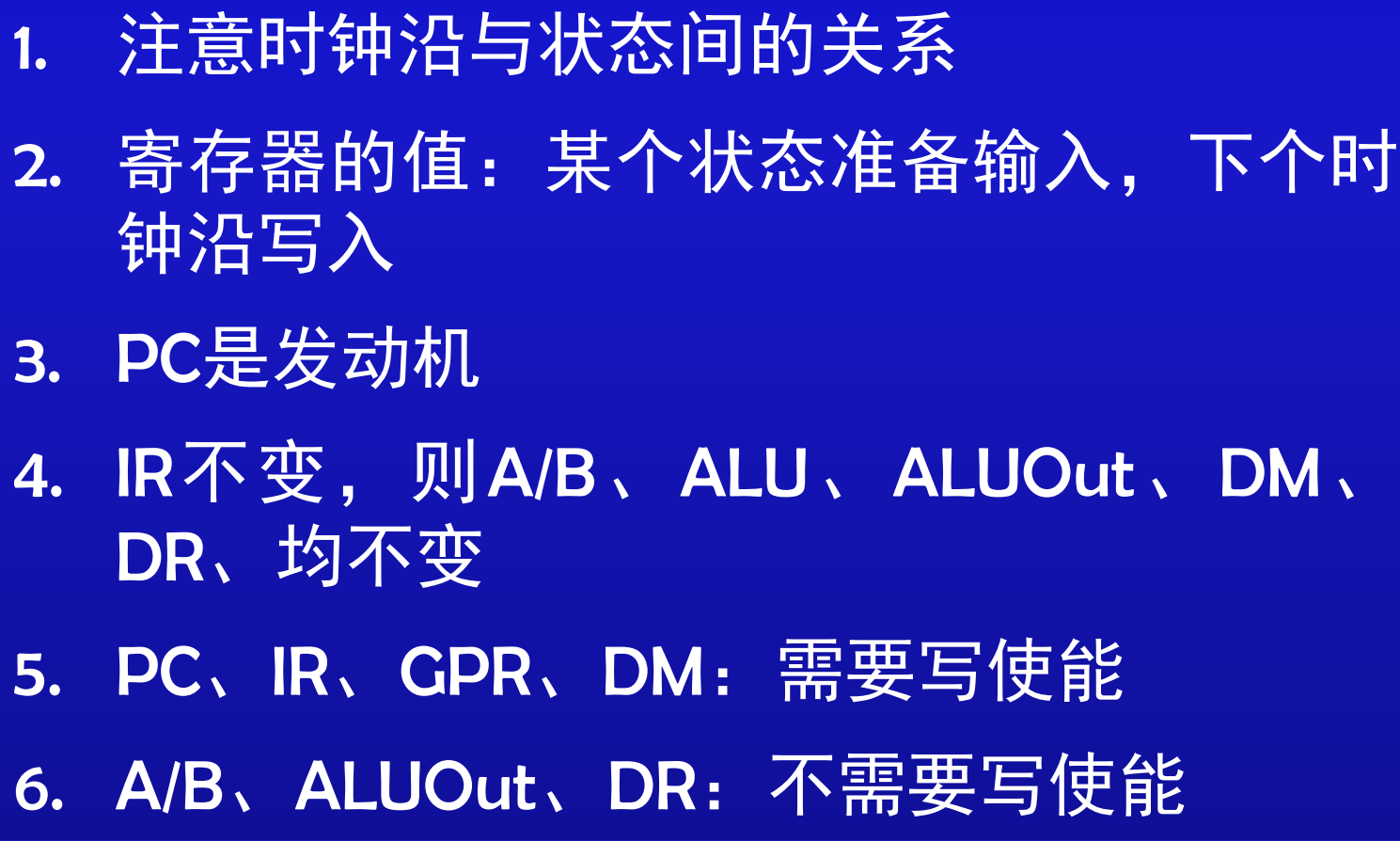
最终 写入

ADDU
SUBU
ORI
LW
SW
BEQ
JAL



	FETCH	DCD/RF	EXE	WB	FETCH
PC	ADDU地址	PC+4	PC+4	PC+4	PC+4
IM	输出ADDU	指令[PC+4]	指令[PC+4]	指令[PC+4]	指令[PC+4]
IR	旧指令	ADDU	ADDU	ADDU	ADDU
A/B	旧操作数	旧操作数	新操作数	新操作数	新操作数
Ext	无效扩展	无效扩展	无效扩展	无效扩展	无效扩展
ALU	执行前指令	执行前指令	计算	计算	计算
ALUOut	前结果	前结果	前结果	计算结果	计算结果
DM	无效数据	无效数据	无效数据	无效数据	无效数据
DR	无效数据	无效数据	无效数据	无效数据	无效数据
GPR	旧值输出	新值输出	新值输出	新值输出	输出/更新

ADDU
SUBU
ORI
LW
SW
BEQ
JAL



FETCH
PC+4
指令[PC+4]
ADDU
新操作数
无效扩展
计算
计算结果
无效数据
无效数据
输出/更新

[illegible]

